

Raising Agents

Chris Darken
Associate Prof., CS
cjdarken@nps.edu

Challenges for Agent Technology

- Individual combatant modeling
- Very novel types of warfare
- Stand-alone trainers
- Modest budgets

Engineering Agents

Agents for military sims are currently **engineered**, i.e. they are **complex** programs written by software/knowledge engineers

Are our engineering techniques up to today's challenges?

Adding even modest increases in capability for complex programs is expensive

Engineered agents tend to be “brittle”, i.e. to break on scenarios slightly different from the test cases

The Grail: Learning Agents

Just “drop them in” the sim and watch them build themselves
Agent learns how the world works by interacting with it
Goals can then be specified and the agent uses its world model to achieve them

Would solve all our problems

Agent builds itself, so we have little development cost

Agent thoroughly explores the environment, so it is not brittle

The most-explored model to date has been that of evolution, (genetic algorithms/programming ,reinforcement learning)

Pro: Can learn everything (universal model)

Con: Much too slow, often does not learn anything useful

Development vs. Evolution

We propose to use the model of **developmental learning** instead of evolution

Developmental learning is the learning that occurs within the lifetime of an individual organism

Therefore, it is always fast compared to evolution

Sometimes it is incredibly fast

- One-shot learning

- Social learning

But it certainly cannot learn any arbitrary task, at least not quickly

What if we build agents with fast, reliable learning of easy patterns instead of universality?

Raising Agents

Unlike universal learning approaches, we cannot expect agents to learn in arbitrary environments

Environments need to be “nurturing”

The way to make a great scientist is not to put children in a lab and turn them loose

This is not merely because they might hurt themselves, but because they will quickly encounter complexities that they will not be able to figure out unaided

Developmental learning is fast, but it requires bite-sized chunks (“learning increments”)

The Problem of Relevance

Assumption: The reason we are able to learn many things quickly is because nature “slow-pitches” them to us

I.e. we can usually see what facts are related to what other facts, and if we cannot we usually fail

Most important examples

Spatial and temporal proximity of cause and effect

Association by similarity

Raising Agents: Prospectus

To raise an agent for some task you would:

Train: Construct learning tasks and experiences making relevance as clear as possible

Model: Build the world containing the learning tasks including adequate perceptual modeling

The cost of knowledge engineering has thus been traded for a world modeling and training cost

This may be a very good trade!

Testbed Virtual Environment

Simple testbed virtual environment built

Small combat-oriented MUD

Networked multiplayer

Explicit discrete-event world model

Events coded as logical atoms

Prototype learning algorithm

Agent acts randomly: focus is prediction of outcome (mental simulation)

Can It Work? Early Results

3 million “percepts” over a calendar week

1,000’s to 100,000’s of predictive “rules”
(actually statistical models) learned

Average predictive capability of about
60% (about 85% on percepts not
intrinsically random)

Future Work

Apply generalization techniques
Exploit predictions

Conclusions: A New Learning Model for Agents

Modeling developmental learning rather than evolution

Cons

- Incapable of universal learning
- Requires “nurturing” environment

Pros

- Extremely fast
- “Nurturing” is fairly intuitive